

目 录

- 软件工程概述
- 软件需求分析
- 系统设计
- 系统实现与程序设计风格
- 系统测试
- 系统实施与维护
- 面向对象的分析和设计方法
- 软件质量及其它

2

CUIT 徐虹



第一章 软件工程概念

软件的特点
软件工程的形成
软件开发模型
软件的生命周期

- 软件的特点
- 软件的分类
 - 按软件的功能划分：系统软件、支撑软件、应用软件和可重用软件
 - 按软件工作的方式划分：实时处理软件、交互式软件和批处理方式
 - 按软件服务对象的范围划分：项目软件和产品软件

6

CUIT 徐虹

软 件 工 程

徐 虹

参考书目

- 软件工程——实践者的研究方法
 - Roger S.Pressman
 - 机械工业出版社
- 软件工程——技术、方法与环境
 - 王立福等
 - 北京大学出版社
- 软件工程方法学及应用
 - 汤庸
 - 中国山峡出版社

3

CUIT 徐虹

软件的特点及分类

- 什么是软件
 - 软件是由计算机系统程序、数据和文档组成。
 - 从广义上将，软件指在研究、开发、维护以及使用上述含义的软件所涉及的理论、方法、技术所构成的学科。

5

CUIT 徐虹

软件的发展和软件危机

- 软件的发展过程
 - 程序设计阶段 50—60年代
 - 面向批处理
 - 有限的分布
 - 自定义软件
 - 程序系统阶段 60—70年代
 - 多用户
 - 实时
 - 数据库
 - 软件产品

- 软件工程阶段（一） 70—90
 - 分布式系统
 - 嵌入
 - 低成本硬件
 - 消费者的影响
- 软件工程阶段（二） 90—今
 - 强大的桌面系统
 - 面向对象的技术
 - 专家系统
 - 人工神经网络
 - 并行计算
 - 网络计算机

- 软件的变化
- 软件危机
 - 很难制定开发计划，开发成本常常高出预算很多，而工作进度缓慢。
 - 软件产品不能满足用户的需求
 - 软件难于维护
 - 由于测试工作不完备使软件的质量较差
 - 重复开发类似的软件
 - 软件生产不能满足社会发展的需要

软件工程

- 要求：人员配合，开发阶段衔接良好，软件接受用户检查。
- 按工程化的原则和方法组织软件开发工作是有意义的，也是摆脱软件危机的一个主要出路。
- 软件工程的定义：运用系统的、规范的和可量化的方法来开发、运行和维护软件

软件工程框架

- 软件工程的目标：生产具有正确性，可用性以及开销合算合宜的产品
- 软件工程的关键元素
 - 方法：提供构造软件的技术
 - 语言：用以支持软件的分析，设计和实现
 - CASE工具：为方法和语言提供自动化或半自动化的支持
 - 过程：将方法，语言和工具黏结在一起，使计算机软件开发理性和适时化。

- 软件工程的基本原则
 - 选取适宜的开发模型
 - 采用合适的设计方法
 - 提供高质量的工程支持
 - 重视开发过程的管理

■ 软件的生命周期

- 计划：制定实施计划
- 需求分析
- 软件设计
- 软件实现
- 确认：运行，测试
- 软件支持：维护



软件开发模型

■ 瀑布模型

- 瀑布模型的原理：将软件生存周期的各项活动规定为依顺序连接的若干阶段，分别依次进行实现。
- 瀑布模型的演变
 - 分析—编码
 - 做什么—如何做—制作—检测—使用
 - 系统需求—软件需求—需求分析—设计—编码—测试—运行
 - 可行性研究——需求分析和规约——设计和规约——编码和单元测试——集成测试和系统测试——交付——维护

■ 瀑布模型的特点

- 工作进程连续规范
- 对每一阶段的工作都需要进行评审
- 缺乏灵活性，不能反映认识过程的反复性
- 适合软件需求非常明确，实际方案确定，编码环境熟悉等所有阶段都有较大把握的软件开发过程。

■ 分析方法：结构化方法（SD, SA, SADT）

- SADT：用系统的思想，系统工程的方法，按用户至上的原则，结构化、模块化，自顶向下对系统进行分析与设计。
- JACKSON方法：适用于一般数据处理系统的结构化方法。
- WARNIER方法：或称逻辑地构造程序的方法（LCP），面向数据结构。

■ SADT的基本思想

- 建立面向用户的观点
- 严格区分工作阶段
- 结构化，模块化，自顶向下进行开发
- 充分预料可能发生的变化：环境、系统和要求
- 工作文件的标准化和文献化

■ 瀑布模型的开发过程

■ 系统需求分析

- 可行性分析10%
- 确定系统开发目标：将可行性研究时提出的目标进一步扩展
 - 系统概貌
 - 开发工作计划
 - 开发所需资源清单
 - 预期效益清单
 - 方案预算

- 需求定义：定义一个可行的，成本效益合理的系统。确定信息的需求量，拟定信息流程。15%
 - 输出格式
 - 输入格式
 - 定义数据项
 - 响应时间，精度，更新次数和数量应予说明

- 系统设计：20%
- 软件开发 25%
- 规程编制 10%
 - 用户说明书
 - 系统管理员使用说明
 - 操作人员使用说明
 - 技术手册
 - 培训资料

- 系统实施 15%
- 运行和维护
- 评审 5%
 - 评价系统效果，形成评审报告
 - 效益，性能，后备和可维护性，改进系统。

■ 原型法

■ 先决条件

- 有快速建立模型的软件工具与环境
- 适用于预先不确定需求的软件开发
- 适用于项目组成员协调，配合，交流困难的情况
- 凭借系统分析人员对用户要求的理解，在强有力的软件环境的支持下，给出系统模型，然后进行评价，根据评价结果修改系统。重复该过程，直至用户满意。

■ 类型

- 抛弃原型法
- 演化原型法
- 增量原型法

■ 模型的评价内容:

- 是否准确反映了信息系统的工作过程
- 是否满足用户要求
- 模型的应用环境及其形式是否合适
- 进一步的要求和改进意见

■ 开发过程

- 确定用户的基本需要, 应用范围, 设计成本, 确定数据
- 开发初始原型: 建立一个交互式系统来满足用户的基本信息需求 强调开发速度, 而非运行效率
- 使用原型系统来确认用户的需求
- 修正和改进原型系统: 产生的运行原型可作为新的应用系统, 或作为应用系统开发的工具

■ 原型法的特点

- 易于接受和掌握, 与人的思维一致
- 以模型为中心
- 准确及时认识问题: 标准统一, 及时反馈意见
- 提高系统效益和适应能力
- 用户参与系统的研制和开发工作
- 易于理解和测试

■ 使用范围

- 不适用的情况
 - 大型系统
 - 大量运算和逻辑性较强的程序模块
 - 管理不善, 信息处理混乱的企业
 - 批处理系统
- 适用的情况

■ 螺旋模型

■ 软件风险

- 产品规模
- 商业影响
- 客户特性
- 过程定义
- 开发环境
- 建造的技术
- 人员的数目及经验

■ 风险因素

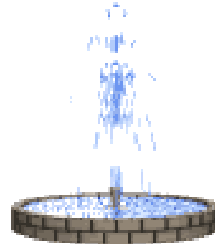
- 性能风险
- 成本风险
- 支持风险
- 进度风险

■ 风险影响的类别

- 可忽略的
- 轻微的
- 严重的
- 灾难的

- 风险预测
- 螺旋模型的原理
- 特点：减少软件开发的~~风险~~，适用于大项目的开发
- 风险缓解、监控和管理（RMMM）

- 面向对象软件开发模型（喷泉模型）



- 软件开发的各个步骤没有明显的界限
主要用于支持面向对象的开发过程。
- 面向对象的方法=对象（属性及服务的封装）
 - +分类
 - +继承
 - +通过消息的通讯

- 四代技术
 - 软件开发环境：数据库查询语言，报表生成，数据处理，屏幕交互和定义，代码生成，高层图形功能和电子表格等。
 - 需求分析----设计策略----用4GL实现--测试

- 其他开发方法
 - 有限状态机方法{FSM}：适用于实时事务处理系统
 - PETRI网方法：适用于并发软件系统
 - 形式化开发方法（FDM）：以数学概念和理论为基础的。

软件工程方法学

- 是指应用计算机科学理论和工程、方法相结合的研究方法，研究软件生存周期的一切活动（包括软件定义，分析，设计，编码，测试与正确性证明，维护与评价等）的方法，工具和管理学的学科。

■ 软件工程方法学的基本内容

- 结构化理论与方法
- 模块技术与数据抽象
- 软件测试与程序正确性证明
- 软件分析与设计方法，工具及环境
- 软件工程管理 with 质量评价

